

# (1) /dev/null Wikipedia is sustained by people like you. Please *donate* today.

From Wikipedia, the free encyclopedia  
(Redirected from Dev/null)

→ In Unix-like operating systems, **/dev/null** or the **null device** is a special file that discards all data written to it (but reports that the write operation succeeded), and provides no data to any process that reads from it (it returns EOF). In Unix programmer jargon, it may also be called the **bit bucket** or **black hole**.

The null device is typically used for disposing of unwanted output streams of a process, or as a convenient empty file for input streams. This is usually done by redirection.

This entity is a common inspiration for technical jargon expressions and metaphors by Unix programmers, e.g. "please send complaints to /dev/null," "my mail got archived in /dev/null," and "redirect to /dev/null," being jocular ways of saying, respectively: "don't bother to send any complaints," "my mail got deleted," and "go to hell." A famous advertisement for the Titanium PowerBook G4 read *The Titanium Powerbook G4 Sends other UNIX boxes to /dev/null*.

The null device is also a favorite subject of technical jokes, such as warning users that the system's /dev/null is already 98% full. The April Fool's, 1995 issue of the German magazine *c't* reported on an enhanced /dev/null chip that would efficiently dispose of the incoming data by converting it to a flicker on an internal glowing LED.

/dev/null is a special *file*, not a directory, so one cannot move files into it with the Unix `mv` command. The `rm` command is the proper way to delete files in Unix.

The equivalent device in CP/M (and later DOS and Windows) is called `NUL:` or just `NUL` (for example, one may hide output by directing it to `NUL`, e.g. `PAUSE>NUL`, which waits for the user to press any key without printing anything to the screen). Under classic Amiga operating systems, the device's name is `NIL:`. In Windows NT and its successors, it is named `\Device\Null` internally, though, the DOS `NUL` is a symbolic link to it. Similarly, in OpenVMS the device is named `NL:`.

## See also

- Unix philosophy
- Standard streams
- /dev/zero
- /dev/random
- /dev
- Device file
- rm (Unix)
- Write Only Memory

## References

- Single Unix Specification Section 10.1
- Jargon file entry for "bit bucket"

Retrieved from "http://en.wikipedia.org/wiki/dev/null"

Categories: Unix | Device file | Nothing

Hidden category: Articles lacking in-text citations

- 
- This page was last modified on 30 June 2008, at 00:01.
  - All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.) Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a U.S. registered 501(c)(3) tax-deductible nonprofit charity.

## (2) NULL

Section: • Linux Programmer's Manual • (4)

Updated: 1992-11-21

[Index](#)

[Return to Main Contents](#)

---

### NAME

null, zero - data sink

### DESCRIPTION

Data written on a **null** or **zero** special file is discarded.

→ Reads from the **null** special file always return end of file, whereas reads from **zero** always return \0 characters.

**null** and **zero** are typically created by:

```
mknod -m 666 /dev/null c 1 3
mknod -m 666 /dev/zero c 1 5
chown root:root /dev/null /dev/zero
```

### NOTES

If these devices are not writable and readable for all users, many programs will act strange.

### FILES

*/dev/null*  
*/dev/zero*

### SEE ALSO

[\*\*chown\*\*\(1\)](#), [\*\*mknod\*\*\(1\)](#)

---

### Index

[NAME](#)

[DESCRIPTION](#)

[NOTES](#)

[FILES](#)

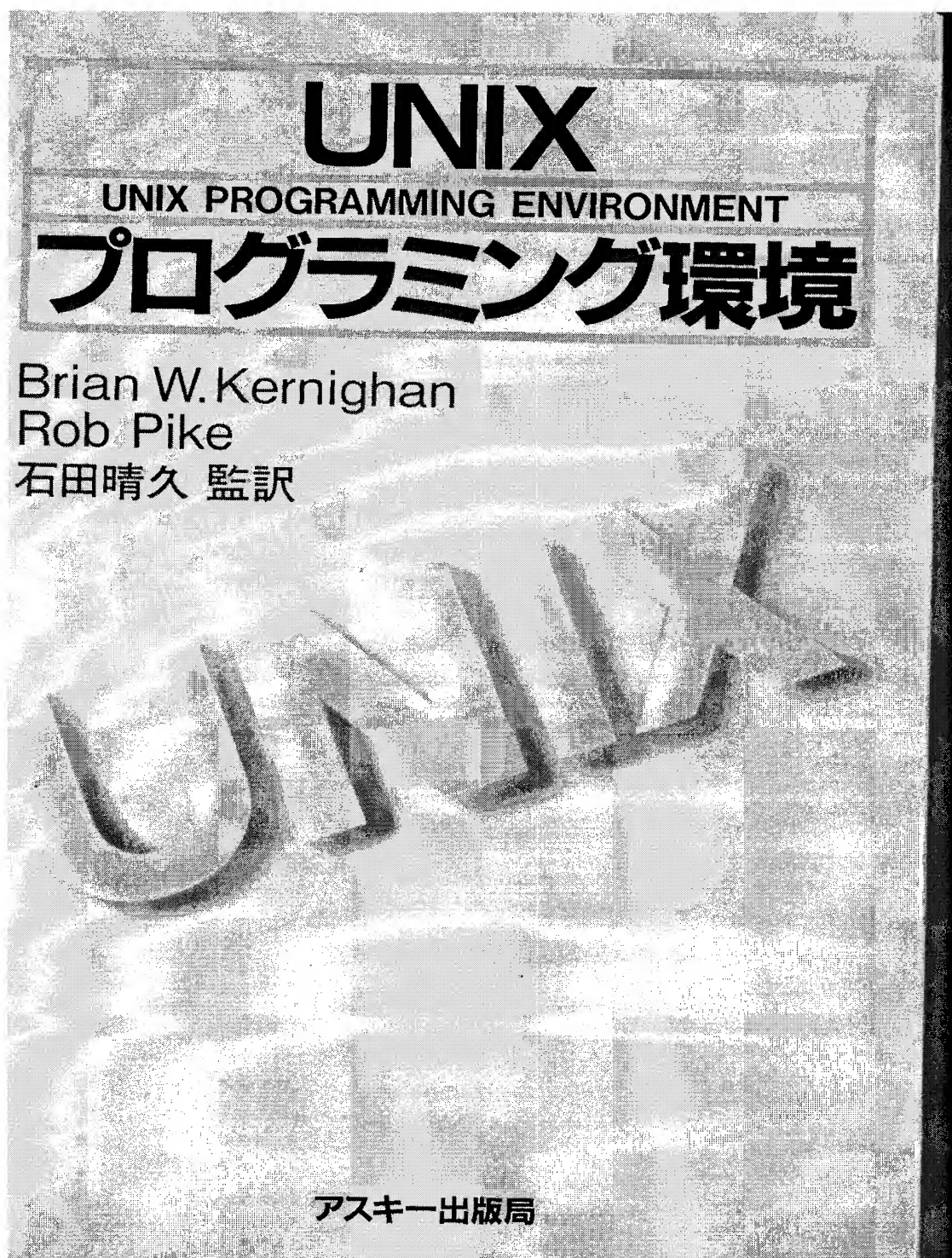
[SEE ALSO](#)

---

This document was created by [man2html](#), using the manual pages.

Time: 04:31:25 GMT, July 01, 2008

(3)



このデバイスの所有者はユーザ (you) で、しかもそのユーザだけが読むことを許可されていることに注意しよう。言い換えれば、他のどんな人にもユーザが入力中の文字を直接には読めないのである。しかし端末への書き込みはどの人にも許されている。これを防ぎたければ、そのデバイスに `chmod` コマンドを実行して、他の人が `write` コマンドを使ってコンタクトをとろうとするのを阻止するわけである。あるいは `mesg` コマンドを使うだけでもよい。

.....メッセージがないようにする。

```
$ mesg n
$ ls -l /dev/tty0
crw-rw---- 1 you
$ mesg y
$
```

1. 12 Sep 28 02:41 /dev/tty0  
.....元に戻す。

使用中の端末を特定の名前で指定できることは役に立つことが多いといえ、いちいち指定するのはやっかいである。そこで、実際に使っている端末に関係なく、ログイン端末と同義語として使えるのが `/dev/tty` というデバイス名である。

```
$ date > /dev/tty
Wed Sep 28 02:42:23 EDT 1983
$
```

この `/dev/tty` は、仮に標準入力と標準出力が端末ではなくファイルと結合されているときでも、あるプログラムがユーザと対話する必要があるときにとりわけ役立つ。`/dev/tty` を使っているプログラムの1つに `crypt` がある。`crypt` の標準入力へ出ていく暗号化される前のテキストが入り、暗号化されたデータは標準出力へ出ていく。そこで `crypt` は、暗号化のためのキーワードを `/dev/tty` から読み込むのである。

```
$ crypt <cleartext >crypttext
Enter key: .....暗号化のためのキーワードをタイプせよ
$
```

`/dev/tty` を使っていることはこの例では目にみえないが、実際に使われている。もし `crypt` が暗号化のキーワードを標準入力から読み込むようになっていれば、暗号化前のテキストの最初の行を読んてしまいうだろう。そこで代わりに、`crypt` は `/dev/tty` をオープンし、暗号化のためのキーワードが画面上でみえないようにするために

の自動エコーバックを止めておいてから、そのキーワードを読み込んでいるのである。第5章と第6章で、`/dev/tty` を利用した別の例を紹介している。

プログラムを走らせてみたいが、その出力は気にしないということとがときにはあるかもしれない。例えば、今日のニュースはもうみてしまったので、もう読みたいくないというような場合、次のように、`news` の出力を `/dev/null` というファイルに切り換えると、その出力が捨てられる。

```
$ news > /dev/null
$
```

`/dev/null` に書き込まれるデータは何の断わりもなく捨てられる。一方、`/dev/null` から読み込むプログラムは、いきなりファイルの終了を受け取ることになる。

この `/dev/null` がよく使われるのは、診断メッセージだけをききみるために、普通の出力を捨ててしまいたい場合である。例をあげると、あるプログラムの使った CPU 時間を教えてくれる `time` コマンド (`time (1)`) の情報はその標準エラー出力に書かれるので、標準出力の方を `/dev/null` に送ってしまえば、本来なら延々と表示するコマンドの CPU 時間もうまく計測できるわけである。

```
$ ls -l /usr/dict/words
-r--r--r-- 1 bin      196513 Jan 20  1979 /usr/dict/words
$ time grep e /usr/dict/words > /dev/null

real    13.0
user     9.0
sys      2.7
$ time egrep e /usr/dict/words > /dev/null

real    8.0
user     3.9
sys      2.8
$
```

出た現われた数字は、それぞれ、経過した実時間、そのプログラムの使った CPU 時間、それにプログラムが走っている間にカーネルが使った CPU 時間を意味している。

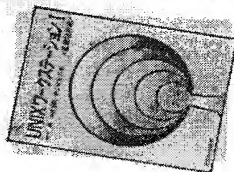
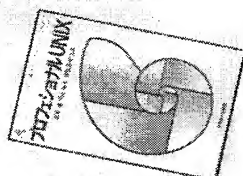


## “UNIX関連書籍”

ソフトウェア エイセス

UNIX  
ワークステーション  
(基礎技術編)村井 純/砂原秀樹/  
横手靖彦 共著A5判 定価2,700円(¥300円)  
(本体2,400円)

〈内容〉

マシン・アーキテクチャ  
オペレーティング・システム  
ネットワーク・アーキテクチャ  
ネットワーク・アプリケーション  
ネットワークにおけるプロトコル  
とデータの表現形式  
ウィンドウシステムソフトウェア エイセス  
プロフェッショナル  
UNIX村井純 井上尚司/砂原秀樹・共著  
A5判 定価2,580円(¥300円)  
(本体2,500円)

〈内容〉

オペレーティング・システムUNIX/UNIXの  
構造/システム環境の設定と管理/生活環  
境の整備/ソフトウェア開発環境/ネット  
ワーク環境/非標準ツール/APPENDIXUNIXの動作メカニズム、システム環境の  
カスタマイズなど、その根底に流れる設計思想  
と実装手法について、解説。

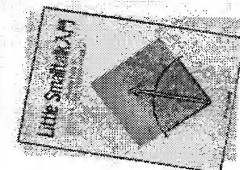
海外ブックス

UNIX  
システムコール・  
プログラミングMaro J. Roehkind 著  
福岡俊博 訳A5判 定価3,610円(¥300円)  
(本体3,500円)

〈内容〉

基本的概念/ファイルI/O (1)/ファイルI/O  
(2)/端末I/O/プロセス/プロセス間通信  
(1)/プロセス間通信(2)/シグナル/その他の  
システムコール/付録A システムVのプロセス  
属性/付録B 標準サブルーチンUNIXのシステムコールを主題に高度なプログラ  
ミングテクニックを解説。

ソフトウェアエイセス

Little Smaltalk  
入門Timothy Budd 著 吉田雄二 監修  
長谷川明生・大田毅勝 訳  
A5判 定価2,680円(¥300円)  
(本体2,600円)

〈内容〉

—入門と言語—  
基礎的事項/構文/基本的なクラス/クラス定義  
/簡単なアプリケーション/カスケード・プリミ  
ティブ・数と型変換 (coerce)/シミュレーション  
/ジェネレータ/グラフィックス/プロセス  
—実現と環境—  
実現の概観/オブジェクトの表現/バイトコード  
/プロセスマネージャ/インタープリタ

\*UNIXオペレーティングシステムはAT&amp;Tのベル研究所が開発し、AT&amp;Tがライセンスしております。

## UNIX プログラミング環境

1985年10月15日 初版発行  
1989年11月11日 第1版第11刷発行  
定価3,910円(本体3,796円)

著者 Brian W. Kernighan &amp; Rob Pike

監訳 石田誠夫

翻訳者 野中浩一

発行者 塚本健一郎

発行者 株式会社アスキー

〒107 東京都港区南青山6-11-1 スリーエフ南青山ビル

電話 東京 4-161144

TEL (03)486-7111(大代表)

情報 TEL (03)498-0299(ダイヤルイン)

出版営業部 TEL (03)486-1977(ダイヤルイン)

本書は著作権法上の保護を受けています。本書の一部あるいは全部  
について(ソフトウェア及びプログラムを含む)を、株式会社アスキー  
から文書による許諾を得ずに、いかなる方法においても無断で複写、  
複製することは禁じられています。

編集担当 佐々木敏久

印刷 モリモト印刷株式会社

C T S 株式会社電算プロセス

ISBN4-87148-351-7 C3055 P3910E

●11108/11379